Learning Chinese Word Embeddings by Discovering Inherent Semantic Relevance in Sub-characters

Wei Lu
National Engineering Research Center for Big Data Technology and System, Services Computing Technology and System Lab, Cluster and Grid Computing Lab
School of Computer Science and Technology, Huazhong University of Science and Technology
Wuhan, China
luwei@hust.edu.cn

Zhaobo Zhang
National Engineering Research Center for Big Data Technology and System, Services Computing Technology and System Lab, Cluster and Grid Computing Lab
School of Computer Science and Technology, Huazhong University of Science and Technology
Wuhan, China
zhang_zb@hust.edu.cn

Pingpeng Yuan
National Engineering Research Center for Big Data Technology and System, Services Computing Technology and System Lab, Cluster and Grid Computing Lab
School of Computer Science and Technology, Huazhong University of Science and Technology
Wuhan, China
ppyuan@hust.edu.cn

Hai Jin
National Engineering Research Center for Big Data Technology and System, Services Computing Technology and System Lab, Cluster and Grid Computing Lab
School of Computer Science and Technology, Huazhong University of Science and Technology
Wuhan, China
hjin@hust.edu.cn

Qiangsheng Hua
National Engineering Research Center for Big Data Technology and System, Services Computing Technology and System Lab, Cluster and Grid Computing Lab
School of Computer Science and Technology, Huazhong University of Science and Technology
Wuhan, China
qshua@hust.edu.cn

ABSTRACT

Learning Chinese word embeddings is important in many tasks of Chinese language information processing, such as entity linking, entity extraction, and knowledge graph. A Chinese word consists of Chinese characters, which can be decomposed into sub-characters (radical, component, stroke, etc). Similar to roots in English words, sub-characters also indicate the origins and basic semantics of Chinese characters. So, many researches follow the approaches designed for learning embeddings of English words to improve Chinese word embeddings. However, some Chinese characters sharing the same sub-characters have different meanings. Furthermore, with more cultural interaction and the popularization of the Internet and web, many neologisms, such as transliterated loanwords and network terms, are emerging, which are only close to the pronunciation of their characters, but far from their semantics. Here, a tripartite weighted graph is proposed to model the semantic relationship among words, characters, and sub-characters, in which the semantic relationship is evaluated according to the Chinese linguistic information. So, the semantic relevance hidden in lower components (sub-characters, characters) can be used to further distinguish the semantics of corresponding higher components (characters, words). Then, the tripartite weighted graph is fed into our Chinese word embedding model *insideCC* to reveal the semantic relationship among different language components, and learn the embeddings of words. Extensive experimental results on multiple corpora and datasets verify that our proposed methods outperform the state-of-the-art counterparts by a significant margin.

CCS CONCEPTS

• **Computing methodologies** → **Lexical semantics**; **Knowledge representation and reasoning**.

KEYWORDS

Natural language processing, Chinese word embeddings, Knowledge processing

ACM Reference Format:
Wei Lu, Zhaobo Zhang, Pingpeng Yuan, Hai Jin, and Qiangsheng Hua. 2022. Learning Chinese Word Embeddings by Discovering Inherent Semantic Relevance in Sub-characters. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM '22), October 17–21, 2022, Atlanta, GA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3511808.3557376

CIKM, October 17-22, 2022, Georgia, Atlanta

ACM ISBN 978-1-4503-9236-5/22/10…$15.00
https://doi.org/10.1145/3511808.3557376

1 INTRODUCTION

Word embeddings, where words or phrases in the corpora are mapped into low-dimension vectors of real numbers, are popular techniques for language modeling [5, 10], feature learning [12], and knowledge processing [21, 25]. For example, the knowledge graph, a network of interconnected knowledge, is constructed by recognizing entities and their relationships or links from free texts, databases and other sources. The correct identification of entities and links is generally based on learning the embeddings of words. From the perspective of linguistics, the aim of learning word embeddings is to capture and learn the semantic relation, interaction, as well as the contextual co-occurrence of words.

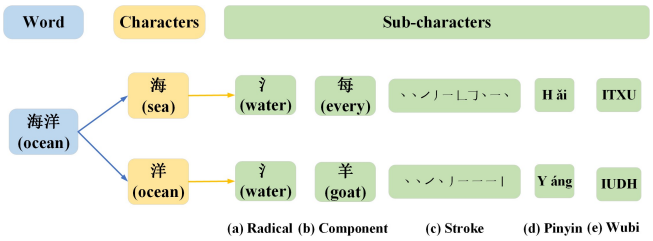


Figure 1: The hierarchy of the Chinese word: word, character, sub-character. The word consists of multiple characters, and the character consists of various sub-characters, including (a) Radical, (b) Component, (c) Stroke, (d) Pinyin, (e) Wubi.

The general approach to learning word embeddings is treating texts as word sequences, which ignores the internal semantic relation and interaction among words. However, Chinese words consist of characters, which further consist of sub-characters–radical, component, etc. For example, "海洋(ocean)" consists of "海(sea)" and "洋(ocean)" in Figure 1. "海(sea)" is decomposed into several sub-characters, such as the radical "氵(water)" and component "每(every)". As the example shows, sub-characters denote partial semantics of words and characters. Therefore, discovering the semantics of sub-characters can improve Chinese word embeddings.

Recently, researchers have attempted to apply sub-characters to learn Chinese word embeddings. For example, JWE [29] decomposed words into sub-characters for refining the semantics of words. Although JWE learns the semantics of words by incorporating sub-characters, the semantics of sub-characters are highly relevant with both Chinese characters and Chinese words. For example, the sub-characters of "杏(apricot)" and "呆(dull)" are both "木(wood)" and "口(mouth)". Obviously, "木(wood)" is more relevant to "杏(apricot)" than "呆(dull)" in its original semantics. However, JWE does not discriminate them and encode the characters using the same sub-characters, causing ambiguity in identifying Chinese characters and words. Besides, the semantics of sub-characters to characters may vary according to the words. For example, the words "海洋(ocean)" and "洋气(fashion)" share the same character "洋(ocean)", but have different meanings (refer to Section 3). This diversity causes the ambiguity in learning Chinese word embeddings. Few researches are reported on discovering such internal semantic interaction among words, characters, and sub-characters.

Furthermore, due to the globalization, many neologisms are introduced into Chinese, such as transliteration loanwords and technical terms. For example, "粉丝(fans)" is a transliteration loanword common in youth. However, it is also a kind of Chinese noodles. For neologisms, both their characters and sub-characters do not relate to them semantically. This case exacerbates the ambiguity in identifying the semantics of Chinese characters and poses new challenges for learning Chinese word embeddings.

In conclusion, recent researches incorporate sub-characters into learning Chinese word embeddings, but ignore the internal semantic relation and interaction among words, characters, and sub-characters. So, the internal semantic relationship can be employed to enhance Chinese word embeddings. In order to address the aforementioned issues, we employ a tripartite weighted graph for managing the inherent semantics of different language components, and propose a Chinese word embedding model *insideCC* for discovering and learning the internal semantic relation and interaction. Extensive experimental results verify the strength of *insideCC* in incorporating the internal semantic information and capturing the contextual co-occurrence information. Our contributions are summarized as follows:

1. The tripartite weighted graph is proposed to model the semantic relationship among words, characters, and sub-characters. The semantic relationship is evaluated according to the linguistic information among related language components. So, the semantic relevance hidden in lower components (sub-characters, characters) can be used to further distinguish the semantics of corresponding higher components (characters, words).

2. The Chinese word embedding model *insideCC*, which accepts the tripartite weighted graph as input, is proposed to enhance Chinese word embeddings. With the tripartite weighted graph, *insideCC* can learn the semantic relation among language components and then strengthen the semantics of words. Since the components of neologisms (transliteration loanwords, etc) have loose semantic relationship among them in the tripartite weighted graph, *insideCC* can identify the neologisms.

3. Extensive experimental results in word similarity and word analogy reasoning tasks verify that *insideCC* outperforms the state-of-the-art counterparts by a significant margin. Quantitative analysis and additional case studies provide sufficient proof of *insideCC*'s semantic learning capability.

2 RELATED WORK

2.1 Semantic methods

Semantics refer to the meaning of the language component, such as word, character, and radical. Strenuous efforts have been made to explore the semantics hidden in different language components. For instance, Chen et al. [3] proposed CWE to improve the representations of Chinese words by incorporating the semantic information of Chinese characters. In order to incorporate the semantic contribution of Chinese characters, SCWE [27] assigned weights to characters by calculating the similarity between English translations , while ACWE [13] employed attention mechanism to perceive the semantic relation between words and characters.

Considering the composition structure of Chinese, there are tens of thousands of words in Chinese, which are decomposed of

thousands of characters. The characters might have difficulties in revealing the semantics inside themselves. So, several researches make a scrutiny into sub-characters in an attempt to understand the meaning of a word. JWE [29] and RECWE [4] enhanced Chinese word embeddings by decomposing characters into sub-characters and utilizing the semantic information of sub-characters. However, these approaches cannot distinguish the characters composed of the same sub-characters, such as "杏(apricot)", and "呆(dull)", whose sub-characters are "木(wood)" and "口(mouth)".

Considering the hierarchy of Chinese words, Yin et al. [28] proposed MGE to learn Chinese word embeddings by utilizing the combination of words, characters, and radicals. Song et al. [22] proposed LSN for jointly learning the semantics by maximizing the overall probability of the relation among words, characters, and components. However, they ignore the semantic interaction among words, characters, and sub-characters. For example, the word "海洋(ocean)", the characters "海(sea)" and "洋(ocean)", and the sub-character "氵(water)" have similar semantics and are relevant to water. The internal semantic relation and interaction can provide supplementary semantics for the word and characters.

2.2 Morphological methods

Chinese is a hieroglyphic language which preserves the morphology and original semantic elements, and the meaning of Chinese characters can be conjectured by their morphological information. The morphological information, such as stroke and glyph, is intuitive to readers since it is visual and conforms to the human spatial imagination. Researches on the morphological information have also achieved progress. For instance, Fasttext [1] utilized the n-gram sequences of English letters to enhance word embeddings. Inspired by Fasttext, CW2VEC [2] leveraged the n-gram sequences of strokes in words by regarding strokes as letters. Nevertheless, the characters with identical stroke sequence, such as "土(soil)", "工(work)" and "士(scholar)", cannot be distinguished by these methods.

Glyphs also retain much morphological information since Chinese characters evolve from pictographs and have different structures. Researchers have focused on glyphs in the past five years. For example, GWE [23] utilized all visual parts of Chinese hieroglyphs to enhance word embeddings. Glyce [16] proposed a semantic representation learning method based on glyphs. However, some unnecessary parts will interfere in distinguishing Chinese characters, burden the training process and consume extra resources.

2.3 Phonetic methods

In order to hear, speak, and read, the phonetic information like pronunciation and speech is inseparable from languages. Generally, one Chinese character may correspond to several pronunciation with different meanings, and one pronunciation may also correspond to multiple Chinese characters, causing ambiguity in distinguishing Chinese characters. Therefore, pinyin is usually combined with other sub-characters to improve the distinguishing accuracy of Chinese characters in recent researches. For example, SSP2VEC proposed by Zhang et al. [30] designed a sub-character feature string by incorporating strokes, pinyin romanization and the structure of the characters, so as to enhance Chinese word embeddings. Later, Sun et al. [24] proposed ChineseBERT, which combined the glyph and pinyin to enhance word embeddings, but ignored their intrinsic semantic information.

2.4 Contextual co-occurrence based methods

Generally, sentences are used in oral communication and written expressions. That is, words need to be understood according to their contexts and speech situation. In order to understand the meanings of words, word sequences and overall statistics are considered into researches. Mikolov et al. [17] first proposed representative word2vec models–CBOW and Skipgram, which obtained word embeddings through the contextual co-occurrence information. However, word2vec only employed the local information of the target words, ignoring the global information between the target and contextual words. Therefore, Pennington et al. [19] proposed Glove to decompose the co-occurrence probabilities matrix for learning the local and global information simultaneously.

Also, one word may have different meanings in different situations. In order to explore the meanings of words, Peters et al. [20] proposed ELMo to learn complex features of words and their changes in different contexts. Inspired by ELMo, BERT [6] replaced few words with masks to strengthen the contextual memory during the training process. Developing from BERT, CharBERT [14] constructed word representations based on sub-words, aiming to perceive characters and refine the semantics of words.

3 TRIPARTITE WEIGHTED GRAPH

A Chinese word consists of several characters, which may appear in other words. Each character denotes partial semantics of the word. Similarly, characters may share sub-characters that indicate common partial semantics of the character. In order to describe the semantic relationship among words, characters, and sub-characters, we propose a tripartite weighted graph defined as followed.

Definition 1. *Let the word set $W = \{w_1, w_2, \cdots, w_n\}$, the character set $C = \{c_1, c_2, \cdots, c_m\}$, and the sub-character set $SC = \{sc_1, sc_2, \cdots, sc_l\}$ in the training corpus D. The tripartite weighted graph $G = (V, E, \mathfrak{W})$ where $V = W \cup C \cup SC$, $E \subseteq (W \times C) \cup (C \times SC)$, and $\mathfrak{W}$ is a weight set. $\forall w_i \in W$, if it contains c_j, there exists an edge between w_i and c_j. Similarly, $\forall c_i \in C$ contains sc_j, an edge joins them. Weights on edges indicate the relationship strength between two nodes (words and characters or characters and sub-characters).*

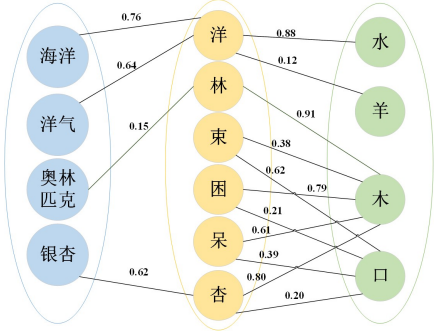


Figure 2: An example of the tripartite weighted graph

Figure 2 is an example of the tripartite weighted graph. The words "海洋(ocean)" and "洋气(fashion)" share the same character "洋(ocean)", which contains the sub-characters "氵(water)" and "羊(goat)". The word "银杏(ginkgo)" contains the character "杏(apricot)". The word "奥林匹克(Olympic)" contains the character "林(forest)", which contains the sub-character "木(wood)". The characters "杏(apricot)", "束(beam)", "呆(dull)", and "困(sleepy)" shares the sub-characters "木(wood)" and "口(mouth)".

Words, characters, and sub-characters in the tripartite weighted graph can be extracted from dictionaries, such as *Modern Chinese Dictionary* [7]. However, the key challenge to build a tripartite weighted graph is to assign weights to edges. Since the semantics of sub-characters are not given in dictionaries and the corpora for sub-characters are rare, it is not straightforward to compute the weights between character and sub-characters. Here, we map sub-characters into characters according to their semantics (Figure 3). For example, the sub-character "氵(water)" is mapped into the character "水(water)". In this way, the corpora for words and characters can be utilized to evaluate the semantics of sub-characters.

sub-characters	characters	sub-characters	characters	sub-characters	characters
糹 纟 系	丝(silk)	犭	犬(dog)	辶	走(walk)
艹 屮 艸	草(grass)	忄	心(heart)	足	足(foot)
釒 钅	金(metal)	亻	人(people)	耂	老(old)
饣 食	食(eat)	灬	火(fire)	牛	牛(cow)
氵 冰	水(water)	月	肉(meat)	衤	衣(cloth)
讠 言	言(speak)	刂	刀(knife)	礻	示(show)
疒	病(illness)	王	玉(jade)	罒	网(net)
扌	手(hand)	冫	冰(ice)	攵	攴(flog)

Figure 3: Mapping sub-characters into characters

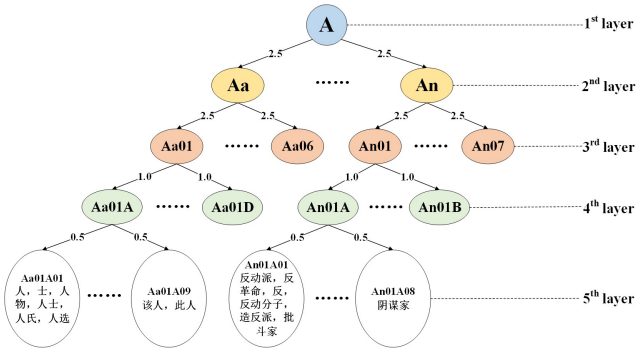


Figure 4: The structure of Tongyici Cilin. The first layer contains 12 major categories (A-L), and the following layers contain the sub-categories of the parent layer. Words and characters are categorized into the clusters with codes like "Aa01A01". Weights on edges denote the lengths of edges.

The semantic relevance of sub-characters varies in different characters. For example, the sub-character "木(wood)" contributes more semantics to "束(beam) "and "杏(apricot)" than "困(sleepy)" and "呆(dull)" (Figure 2). So, it is not feasible to set the same semantic relevance for the same sub-character in different characters. Here, Tongyici Cilin [15], is employed to evaluate the semantic relevance. Similar to WordNet [18], Tongyici Cilin is a tree-shape structure (Figure 4), where non-leaf nodes are the categories, and leaves are the clusters.

Given a character c_i and a sub-character sc_j, we first map sc_j into its equivalent character $char(sc_j)$, and know the locations and categories that c_i and $char(sc_j)$ belong to. According to the locations of c_i and $char(sc_j)$, we can evaluate the semantic relevance by the distances between them and their parents. Inspired by the method proposed by Zhu et al. [31], we further employ *Modern Chinese Dictionary* to collect the co-occurrence of characters since Tongyici Cilin only considers the semantic relevance in isolated characters. Now, the semantic relevance between c_i and sc_j is defined as:

$$sim(c_i, sc_j) = \begin{cases} 0\,, \ sc_j \notin c_i \ || \ char(sc_j) \ or \ c_i \ out \ of \ vocab \\ \frac{\sum w_k \ni sc_j}{|W_{c_i}|}\,, \ only \ char(sc_j) \ or \ c_i \ in \ a \ cluster \\ 1.0\,, \ char(sc_j) = c_i \\ (1.05 - 0.05 * d) * \sqrt{e^{-\frac{g}{2n}}} + \frac{\sum w_k \ni sc_j}{|W_{c_i}|}, others \end{cases} \quad (1)$$

where d is the distance between c_i and $char(sc_j)$, g is the distance between their parents, and n is the children number of their common ancestor since the semantics of the descendants have large divergence if the ancestor has many children. $\sum w_k \ni sc_j$ denotes the amount of the words that contains the characters with sc_j except c_i, and $|W_{c_i}|$ denotes the amount of the words containing c_i.

The semantic relevance can not be used as the weight directly since it may be over-large or over-small in some cases. Many Chinese characters are phono-semantic compound characters. For example, "羊(goat)" is the phonetic component of "洋(ocean)" while "氵(water)" is the semantic component. So, the semantic relevance between "洋(ocean)" and its sub-character "氵(water)" is much greater than the semantic relevance between "洋(ocean)" and its sub-character "羊(goat)". Here, we normalize the semantic relevance between a character and its sub-characters. That is, the weight between character c_i and sub-character sc_j is calculated as:

$$\omega_{i,j} = softmax(sim(c_i, sc_j)) \quad (2)$$

We also utilize Tongyici Cilin and the corpora for words and characters to evaluate the weights between words and characters. For example, the word "奥林匹克(Olympic)" and the character "林(forest)" are far away in Tongyici Cilin, and their contextual co-occurrences are rare in the corpora, so the normalized weight of "林(forest)" to "奥林匹克(Olympic)" is small. Above all, the weight between word w_i and c_j is defined as:

$$\lambda_{i,j} = softmax(sim(w_i, c_j)) \quad (3)$$

4 *INSIDECC*

As the tripartite weighted graph shows, the semantics of words can be indicated at both the character level and sub-character level in some sense. For example, the sub-character "木(wood)" is close to the semantics of the character "林(forest)", but far to the semantics of the transliterated loanword "奥林匹克(Olympic)" because the sub-characters or characters of "奥林匹克(Olympic)" do not share similar semantics with the word. Besides, neologisms, e.g. name ("马云(Jack Ma)") and transliteration loanword ("奥林匹克(Olympic)"), can not be considered as the semantic composition of characters. It motivates us to utilize the semantic association among words and language components to enhance word embeddings.

Here, we propose Chinese word embedding model *insideCC* (Figure 5), which discovers and learns the semantic relation and interaction among words, characters, and sub-characters with the tripartite weighted graph as the input. *InsideCC* combines the embeddings of different language components to generate the ultimate target word embedding and predict contextual words.

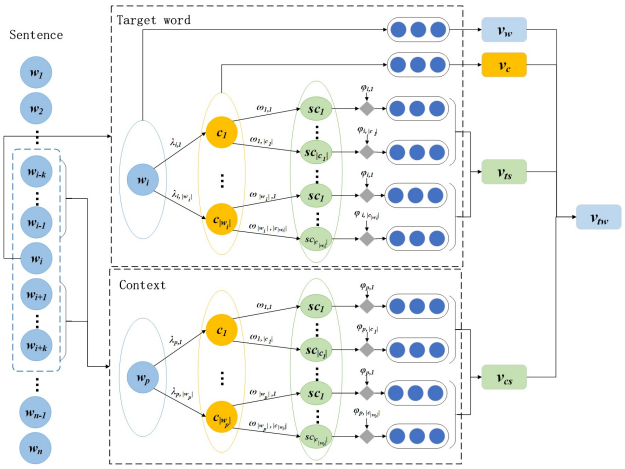


Figure 5: The architecture of Skipgram-based *insideCC* model. w_i represents the target word. w_{i-k} to w_{i-1} and w_{i+1} to w_{i+k} are the contextual words of w_i, which are replaced by w_p in the *Context* square for concise demonstration. c_1 to $c_{|w_i|}$ denote the characters in w_i. sc_1 to $sc_{|c_i|}$ denote the sub-characters in c_i. $\lambda_{i,j}$, $\omega_{i,j}$, and $\varphi_{i,j}$ are the semantic relevance between w_i and c_j, c_i and sc_j, w_i and sc_j, respectively. v_w, v_c, and v_{ts} are the embeddings of the target word, characters, and sub-characters. v_{cs} is the embedding of contextual sub-characters. v_{tw} denotes the ultimate target word embedding.

Algorithm 1 Distinguishing semantic relevance

Input: tripartite weighted graph $G = (V, E, \mathfrak{W})$ where $V = W \cup C \cup SC$
Output: Semantically relevant language components

1: Initialize Queue $Queue_c$ and $Queue_{sc}$
2: **for** word $w_i \in W$ **do**
3: **for** character c_j in w_i **do**
4: $\lambda_{i,j} = softmax(sim(w_i, c_j))$;
5: **if** $\lambda_{i,j} \geq \theta$ **then** ▷ θ is a hyper-parameter
6: Pile c_j and $\lambda_{i,j}$ to $Queue_c$;
7: **for** sub-character sc_k in c_j **do**
8: $\omega_{j,k} = softmax(sim(c_j, sc_k))$;
9: $\varphi_{i,k} = softmax(sim(w_i, sc_k))$;
10: **if** $|\omega_{j,k} - \varphi_{i,k}| \leq \sigma$ **then** ▷ σ is a hyper-parameter
11: Pile sc_k and $\omega_{j,k}$ to $Queue_{sc}$;
12: **return** $Queue_c$ and $Queue_{sc}$;

4.1 Distinguishing semantic relevance

In order to utilize the components to enhance word embeddings, we first need to distinguish the semantic relevance between words and their components. Here, we introduce two hyper-parameters θ and σ that denote the semantic threshold of characters and sub-characters. By comparing the weights between components with θ and σ, we can exclude the characters and sub-characters that are less relevant to words (Algorithm 1). In the algorithm, we first distinguish the semantically relevant characters (Line 2-6), and then identify the semantically relevant sub-characters (Line 7-11).

As Algorithm 1 shows, the semantic relevance can be distinguished with the semantic thresholds of characters and sub-characters, so that the interference of semantically irrelevant characters and sub-characters can be reduced. Thus, the semantics of words will be improved. For instance, since the character "林(forest)" and the sub-character "木(wood)" have little semantic contribution to the word "奥林匹克(Olympic)", the semantic relevance of "林(forest)" and "木(wood)" to "奥林匹克(Olympic)" will be ignored when the tripartite weighted graph is encoded as input.

4.2 Integrating sub-character semantics

Now, the embeddings of the target word, target characters, target sub-characters, and contextual sub-characters will be combined together to generate new target word embedding.

Let v_w, v_c, v_{ts}, v_{cs} represent the embedding of the target word, target characters, target sub-characters, and contextual sub-characters, respectively. Assume there are $|w_i|$ characters in word w_i. The word embedding v_w and character embedding v_t are calculated as:

$$v_w = \hat{e}_{w_i} \tag{4}$$

$$v_c = \frac{1}{|w_i|} \sum_{j=1}^{|w_i|} \frac{1}{2}(\hat{e}_{c_j} + \hat{e}_{\lambda_{i,j}}) \tag{5}$$

where $\hat{e}_{w_i}$, $\hat{e}_{c_j}$, and $\hat{e}_{\lambda_{i,j}}$ are the embeddings of word w_i, character c_j, and the semantic relevance $\lambda_{i,j}$, respectively.

Assume there are S_{w_i} sub-characters in word w_i, A_{w_i} characters in the context, and $|c_u|$ sub-characters in character c_u. The target sub-character embedding v_{ts} and contextual sub-character embedding v_{cs} are calculated as:

$$v_{ts} = \frac{1}{S_{w_i}} \sum_{u=1}^{|w_i|} \sum_{v=1}^{|c_u|} \frac{1}{2}(\hat{e}_{sc_v} + \hat{e}_{\omega_{u,v}}) \tag{6}$$

$$v_{cs} = \frac{1}{A_{w_i}} \sum_{p=i-k}^{i+k} \sum_{u=1}^{|w_p|} \sum_{v=1}^{|c_u|} \frac{1}{2}(\hat{e}_{sc_v} + \hat{e}_{\omega_{u,v}}), \quad p \neq i \tag{7}$$

where $\hat{e}_{sc_v}$ and $e_{\omega_{u,v}}$ are the embeddings of sub-character sc_v and the semantic relevance $\omega_{u,v}$, respectively.

After obtaining the embeddings above, the ultimate target word embedding v_{tw} is calculated as:

$$v_{tw} = \frac{1}{4} \sum_{m} v_m, \quad m = w, c, ts, cs \tag{8}$$

The methods of optimization and negative sampling are the same as those in Skipgram. As for the maximum overall log-likelihood function, *insideCC* employs four conditional probabilities, which are calculated as:

$$L(W) = \sum_{m} \frac{1}{|W|} \sum_{i=1}^{|W|} \sum_{|j| \leqslant k, j \neq 0} logP(w_{i+j}|v_m), \ m = w, c, ts, cs \tag{9}$$

$$P(w_u|v_m) = \frac{exp(\hat{e}_{w_u}^T v_m)}{\sum_{t=1}^{|W|} exp(\hat{e}_{w_t}^T v_m)}, \quad m = w, c, ts, cs \tag{10}$$

5 EXPERIMENTS AND ANALYSIS

5.1 Experimental Settings

5.1.1 Training corpora and evaluating datasets. The training corpora and evaluating datasets are as follows:

- The adopted training corpora are ***Chinese Wikipedia Dump***[1] with 270,292,772 tokens and 766,723 words and ***THUCNews***[2] with 309,526,604 tokens and 397,871 words.
- ***WS240***, ***WS297***, and ***SL999*** [9] are adopted for word similarity tasks, which contain 240, 297, and 999 word pairs along with manually tagged similarity scores, respectively. Since *SL999* is an English dataset, we translate it into Chinese by selecting the most common meaning of each English word.
- ***WA1124*** and ***WA7636*** [11] are used for word analogy reasoning tasks. *WA1124* consists of 677 groups of *capital-belong-country*, 175 groups of *state-include-city*, and 272 groups of *family relations*. *WA7636* consists of 3192 groups of *geography*, 1465 groups of *history*, 1370 groups of *nature*, and 1609 groups of *people*.

5.1.2 Baselines. Our *insideCC* is evaluated against the state-of-the-art models listed below.

- **CBOW** and **Skipgram**[3] [17] are efficient contextual co-occurrence based models, and are widely used as the competitors.
- **CWE**[4] [3] and **SCWE**[5] [27] are character-based Chinese word embedding models. CWE utilizes the semantic information of characters, while SCWE further incorporates the semantic similarities between characters.
- **GWE**[6] [23] and **CW2VEC**[7] [2] are morphology-based Chinese word embedding models. GWE utilizes the glyphs of characters, while CW2VEC utilizes the n-gram sequences of strokes.
- **JWE**[8] [29] is a sub-character based Chinese word embedding model, and utilizes the semantic information of sub-characters.

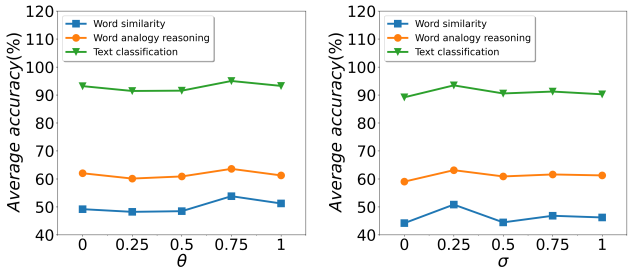
5.1.3 Parameter settings. The **embedding dimension** is 200, the length of the **context window** is 8, the **original learning rate** is 0.025, and the number of **iteration rounds** is 100. As for the negative sampling process, the number of **negative samples** is 10, and the **random negative sampling rate** is 10^{-4}. Furthermore, we conduct parameter analysis to determine the value of the hyper-parameters (Figure 6). We set θ to be 0.75 and σ to be 0.25 because *InsideCC* performs best when $\theta = 0.75$ and $\sigma = 0.25$.

For fair comparison, we keep the common configurations and parameters identical for all models, and adopt other configurations and parameters with the best performance in their papers, as well as adopt average scores to report the final results of multiple tasks.

5.2 Word similarity

Word similarity tasks mainly evaluate the capability of word embeddings to reveal the semantic relation between two words. The results on word similarity tasks (Table 1) demonstrate that *insideCC* outperforms the baselines in the corpora and evaluating datasets.

[1]https://dumps.wikimedia.org/zhwiki
[2]http://thuctc.thunlp.org/
[3]https://code.google.com/p/word2vec
[4]https://github.com/Leonard-Xu/CWE
[5]https://github.com/JianXu123/SCWE
[6]https://github.com/ray1007/GWE
[7]https://github.com/bamtercelboo/cw2vec
[8]https://github.com/HKUST-KnowComp/JWE



(a) Parameter analysis for θ (b) Parameter analysis for σ

Figure 6: Parameter analysis for hyper-parameters

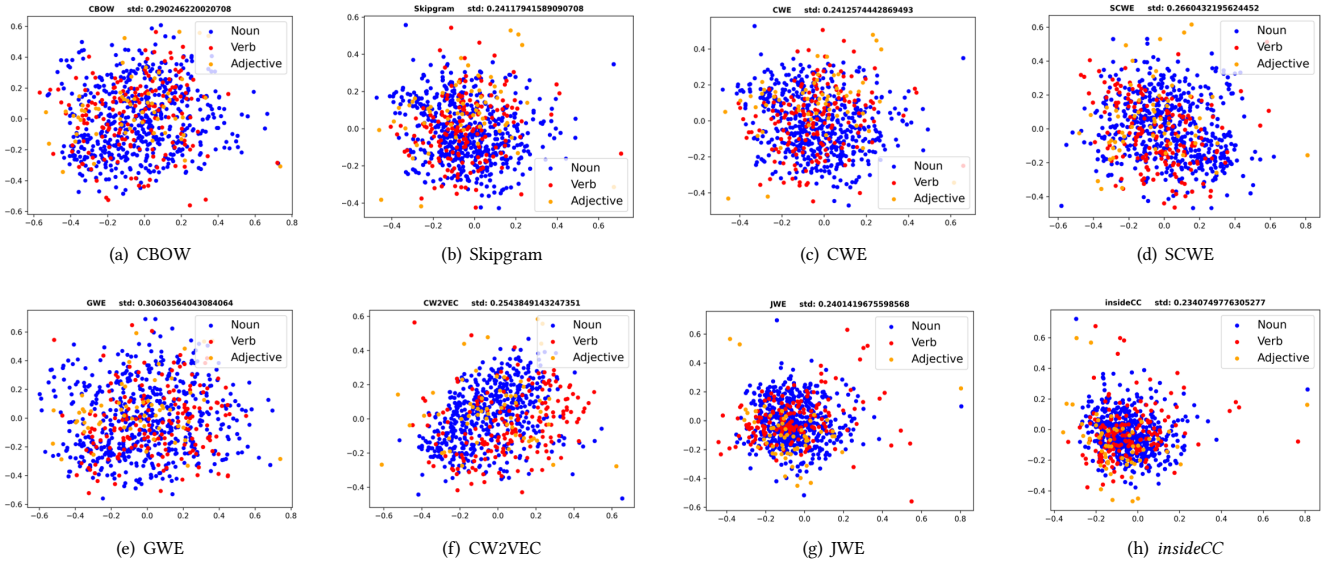
The reason is that *insideCC* accepts the tripartite weighted graph as its input. So, it can learn the semantics of words, characters, and sub-characters, thus achieving better performance than CBOW and Skip-gram. By leveraging additional semantic information of fine-grained sub-characters, *insideCC* also surpasses CWE and SCWE. Compared with JWE, *insideCC* integrates the semantic relevance and incorporates the semantic relation and interaction among words, characters, and sub-caracters, which are not reflected in the baselines. As for GWE and CW2VEC, identical character glyphs and stroke n-gram sequences may bias the training process for learning the semantics of words. Thus, *insideCC* can capture and utilize more semantic information to enhance Chinese word embeddings, and outperform GWE, CW2VEC, and JWE as well.

The results in *SL999* are not better than the results in *WS240* and *WS297*. The reason is that *SL999* is a large dataset and has some translation errors when mapping some English words into Chinese words. For example, the Chinese mapping of the word pair "old, new" in *SL999* is "老(old), 新(new)". However, the best translation should be "旧(old), 新(new)"–a pair of antonyms, which is more reasonable in Chinese. Consequently, the translation errors lead to insufficient semantic information and lower scores in *SL999*.

In order to evaluate the performance of distributing word embeddings in the embedding space, we merge three datasets: *WS240*, *WS297*, and *SL999* and visualize the distribution of all nouns, verbs, and adjectives for eight models (Figure 7). The distribution of CBOW (Figure 7(a)), SCWE (Figure 7(d)), and GWE (Figure 7(e)) is sparser than other models, leading to loose connection among words and lower scores in word similarity tasks. Besides, the distribution of Skipgram (Figure 7(b)), CWE (Figure 7(c)), and CW2VEC (Figure 7(f)) is divided into dense and sparse regions, making the word distribution more imbalanced. For JWE (Figure 7(g)) and *insideCC* (Figure 7(h)), their word distribution is denser than other models because the sub-characters enhance the semantic relation among words. And due to additional utilization of the semantic relevance, *insideCC* has denser word distribution and less isolated words than JWE. Therefore, *insideCC* has the best capabilities for semantic generalization and word distribution. On one hand, the overall standard deviation of *insideCC* is the smallest among the eight models, indicating the word embeddings generated by *insideCC* are more stable in the continuous embedding space. On the other hand, the semantic relation and word connection in *insideCC* are tighter than other models. The reason is that the sub-characters

Table 1: Experiment results (%) of word similarity and word analogy reasoning tasks

Model	*Wikipedia*					*THUCNews*				
	word similarity			word analogy reasoning		word similarity			word analogy reasoning	
	WS240	*WS297*	*SL999*	*WA1124*	*WA7636*	*WS240*	*WS297*	*SL999*	*WA1124*	*WA7636*
CBOW	54.05	58.86	32.08	84.15	39.92	55.74	59.76	31.16	72.69	34.22
Skipgram	56.22	60.51	30.82	84.21	41.35	57.98	59.44	33.45	74.59	39.69
CWE	56.36	62.17	32.42	84.17	41.28	56.92	59.55	33.45	72.76	39.77
SCWE	55.32	58.14	36.95	80.87	35.56	56.23	58.75	35.17	70.58	30.56
GWE	56.02	61.22	31.43	83.73	40.56	52.34	56.63	33.28	71.12	37.16
CW2VEC	55.83	60.25	30.12	80.65	38.34	52.45	59.76	32.62	70.75	36.19
JWE	54.95	64.16	37.25	84.23	39.28	55.37	60.47	37.62	73.75	37.12
insideCC	**56.76**	**65.04**	**39.64**	**85.55**	**41.63**	**59.21**	**61.26**	**38.35**	**81.44**	**40.97**



(a) CBOW (b) Skipgram (c) CWE (d) SCWE

(e) GWE (f) CW2VEC (g) JWE (h) *insideCC*

Figure 7: The distribution of all nouns (blue), verbs(red), and adjectives (orange) of three data sets: *WS240*, *WS297*, and *SL999*. The overall standard deviation is shown at the top of each figure.

reveal relevant semantics and the semantic relevance conveys the semantic contribution of the sub-characters.

The characteristics of the datasets also confirm that *insideCC* can capture more semantic relevance among multiple-granularity language components. For example, nouns like entity names occupy 76.8% in *WS240* and 74.5% in *WS297*, which are 13.1% more than in *SL999* (61.4%). But adjectives and verbs occupy more in *SL999* (8.2%, 22.0%) than in *WS240* (2.2%, 12.4%) and *WS297* (1.8%, 13.1%). The internal differences in the proportion of word properties cause varied performance for 8 models in 3 datasets. The models good at capturing the semantics of nouns may perform better in *WS240* and *WS297*, while those good at learning the semantics of verbs and adjectives may perform better in *SL999*. Further, the proportion of the word pairs with at least one common character and sub-character is 8.75% and 32.5% in *WS240*, 14.48% and 32.99% in *WS297* as well as 17.42% and 38.64% in *SL999*. The proportion distribution is consistent with the improvement in the performance of *insideCC* in word similarity tasks. The matching scores in *WS297* are higher than the scores in *WS240* since the semantic relation is closer in *WS297*. Although there are more semantically relevant word pairs in *SL999*, the matching scores are lower than other datasets due to its larger size and translation errors.

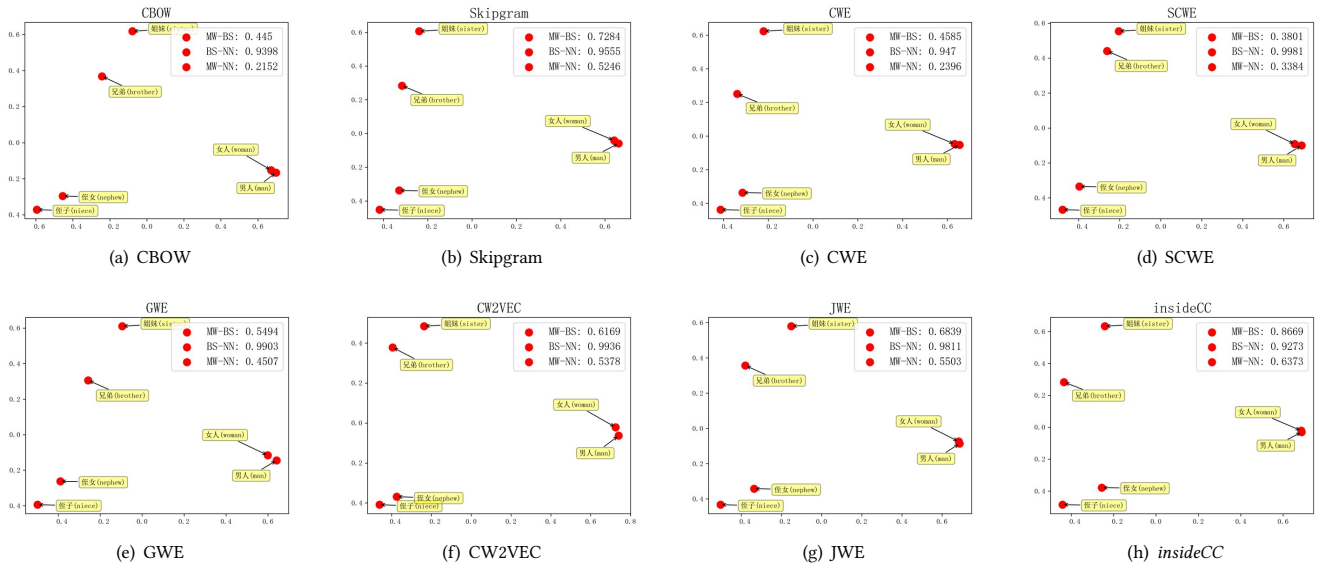
5.3 Word analogy reasoning

Word analogy reasoning tasks aim to evaluate the semantic relevance, contextual association, and linguistic regularities between two pairs of word embeddings. Two datasets are chosen for the evaluation. As mentioned in Section 5.1.1, *WA1124* concentrates more on the semantic information and contextual co-occurrence information, both of which can be captured by *insideCC*. *WA7636* contains abundant neologisms, such as person names, transliterated loanwords, festivals, etc. So, *WA7636* is mainly designed for the linguistic regularities rather than the semantic information.

According to the experimental results (Table 1), *insideCC* outperforms the baselines in all word analogy reasoning tasks. The reason why Skipgram and CWE surpass JWE in *WA7636* is that

Table 2: Analogy details (%) in different datasets and corpora. The score represents the proportion of successful analogy reasoning cases. *cbc*, *sic*, and *fr* represent *capital-belong-country*, *state-include-city*, and *family relation*, respectively.

Model	*Wikipedia*							*THUCNews*						
	WA1124			*WA7636*				*WA1124*			*WA7636*			
	cbc	*sic*	*fr*	*geography*	*nature*	*history*	*people*	*cbc*	*sic*	*fr*	*geography*	*nature*	*history*	*people*
CBOW	89.66	95.43	62.87	55.15	**29.34**	20.08	28.56	74.89	72.00	69.85	41.79	19.58	2.40	35.78
Skipgram	90.39	95.43	59.56	56.15	26.96	23.78	**33.90**	75.92	87.43	62.87	49.29	23.38	0	36.23
CWE	90.99	94.29	58.46	**56.82**	26.49	25.62	33.01	71.64	86.86	62.13	48.59	22.62	0	34.73
SCWE	86.71	90.86	59.93	50.95	26.88	12.81	23.73	69.72	80.00	57.35	34.72	**24.91**	0.80	29.19
GWE	84.19	89.71	**63.24**	55.75	27.84	24.44	29.35	73.86	80.00	57.36	45.95	22.53	0.80	33.53
CW2VEC	87.89	89.14	56.99	51.88	25.30	23.78	29.08	73.86	66.86	65.44	44.15	20.91	1.60	36.38
JWE	91.14	96.00	62.50	56.15	27.91	20.34	24.35	76.51	77.71	64.34	45.76	22.53	1.60	33.83
insideCC	**91.43**	**97.14**	**63.24**	56.45	27.44	**28.53**	30.87	**84.79**	**88.00**	**70.22**	**50.55**	23.57	**6.40**	**38.23**



(a) CBOW (b) Skipgram (c) CWE (d) SCWE

(e) GWE (f) CW2VEC (g) JWE (h) *insideCC*

Figure 8: The visualizations of the Chinese word embeddings. The selected word pairs are ("男人(man)", "女人(woman)")(MW), ("侄子(nephew)", 侄女(niece)")(NN), and ("兄弟(brother), 姐妹(sister)")(BS). Red circles represent words. The cosine similarities between the word embeddings of the selected word pairs are shown at the top right corner of each figure.

the sub-characters of neologisms provide negative semantics while the words and characters reveal positive semantics. Besides, SCWE performs worse than other baselines. The main cause is that the semantic and linguistic relation among words in *WA7636* may change when translating neologisms into English. In addition, *insideCC* can distinguish these neologisms by utilizing the semantic relevance, while the baselines ignore the internal semantic connection and interaction among words, characters, and sub-characters. Therefore, *insideCC* can surpass the baselines.

For more in-depth analysis, we present key details about Word analogy reasoning tasks on different types of word analogies in Table 2. The scores are higher in *WA1124* than *WA7636* due to its smaller dataset size and higher hit rates in each category. *insideCC* achieves the highest hit rates in all categories of *WA1124*. For *WA7636*, *insideCC* performs competitively in *geography*, *nature*, and *people*, and achieves the highest hit rates in *history*.

We choose the words in *family relation* for further analysis because their semantic relationships always keep unchanged and they have high contextual co-occurrence in the corpora. Meanwhile, their characters or sub-characters can extract more commonalities and reveal the semantic relevance, which depends on the distance between words in the embedding space. For instance, "男人(man)" - "女人(woman)" ≈ "侄子(nephew)" - "侄女(niece)". The common characters "人(human)" and "侄(niece)" as well as the common sub-character "亻(human)" can reveal the semantic relation between words. However, the analogy results of "兄弟(brother)" and "姐妹(sister)" may be different since there is no common character or sub-character between them. For better illustration, we demonstrate the visualizations of Chinese word embeddings through PCA (*Principal Component Analysis*) downscaling operation in Figure 8.

Both CBOW and Skipgram incorporate the word co-occurrence. So, Figure 8(a) and 8(b) show that the relative positions of words are quite close. However, they use different prediction methods for learning word embeddings. Thus, the word embeddings of the same words learnt by CBOW and Skipgram separately may be different. In the figures, the absolute positions of words differ a lot.

In Figure 8(c) and 8(d), the distance between "兄弟(brother)" and "姐妹(sister)" varies a lot. CWE utilizes characters to capture the semantic information, so the distance in CWE is longer since there are no common characters in "兄弟(brother)" and "姐妹(sister)". SCWE translates Chinese into English to calculate the semantic similarities of characters. The English words of "兄", "弟" and "兄弟" are "brother", and the English words of "姐", "妹" and "姐妹" are "sister", so SCWE can capture more semantics and relate the words more closely, leading to smaller distance.

For GWE (Figure 8(e)) and CW2VEC (Figure 8(f)), the distance between "男人(man)" and "女人(woman)" becomes longer, indicating that the morphological information may reduce overlapping semantics. Besides, the distances of "兄弟(brother)" and "姐妹(sister)" as well as "侄子(nephew)" and "侄女(niece)" are smaller in CW2VEC due to some identical parts in stroke n-gram sequences, while GWE utilizes the glyphs of characters, which are different in "兄弟(brother)" and "姐妹(sister)", thus leading to larger distance between "兄弟(brother)" and "姐妹(sister)".

In Figure 8(g) and 8(h), the positions of "侄子(nephew)" and "侄女(niece)" are quite close since JWE and *insideCC* both utilize sub-characters for additional supplementary semantic. Besides, compared with JWE, *insideCC* utilizes the semantic relevance to convey more accurate semantic contribution of sub-characters and distinguish the semantics of words, making "姐妹(sister)" farther from "兄弟(brother)" in the embedding space.

More intuitively, the cosine similarities of word pairs indicate the analogy performance. It can be observed that *insideCC* achieves the most accurate results on *MW-BS* and *MW-NN*, which corresponds to the real semantic relations. As for *BS-NN*, it is hard to identify their family relations since "兄弟(brother)" and "姐妹(sister)" have multiple meanings in Chinese, such as blood brothers and sisters, cousins, friends, and even strangers. So, it is unreasonable to regard the relation between "侄子(nephew)" and "侄女(niece)" and the relation between "兄弟(brother)" and "姐妹(sister)" as the same relation directly. For the word pair "兄弟(brother)" and "姐妹(sister)", *insideCC* captures and incorporates more semantics like blood brothers and sisters rather than cousins, thus causing inferior cosine similarity of *BS-NN* to the baselines.

5.4 Case studies

We also perform case studies to assess the capability of *insideCC* in capturing and learning the semantic relation and interaction among words, characters, and sub-characters. Table 3 shows the top 10 closest words to "海洋(ocean)", which are predicted by Skipgram, CWE, JWE, and *insideCC*, respectively.

The words like "极地(polar region)" and "大气(atmosphere)" are semantically irrelevant to "海洋(ocean)", but share some contextual co-occurrence with it, so they are predicted by Skipgram. Besides, most words predicted by CWE are semantically relevant to "海洋(ocean)", except "极地(polar region)" and "大陆架(continental shelf)", indicating that incorporating the semantics of characters is not enough to reveal the complete semantics of words. Even though JWE predicts "大气(atmosphere)", it can associate more words with "海洋(ocean)" than Skipgram and CWE since it integrates the semantics of sub-characters. The reason why JWE associates "大气(atmosphere)" with "海洋(ocean)" is that "大气(atmosphere)" usually occurs with "大海(sea)" and shares the same character "大(large)" with it, resulting in the indirect relation between "大气(atmosphere)" and "海洋(ocean)". The words predicted by *insideCC* all contain the character "海(sea)", and are semantically relevant to "海洋(ocean)". So, *insideCC* can discover the semantic relevance and provide supplementary semantics for sub-characters. Meanwhile, some words such as "海洋资源(ocean resources)" usually occur together with "海洋(ocean)" in the contexts. It shows that *insideCC* can also take advantage of the external contextual co-occurrence information. Therefore, *insideCC* can utilize the internal semantic information and external contextual co-occurrence information to associate words, characters, and sub-characters with relevant semantics, thus making the word embeddings more accurate and representative than other models.

Table 3: The top 10 closest words to "海洋(ocean)" predicted by Skipgram, CWE, JWE, and *insideCC*

Skipgram	CWE	JWE	*insideCC*
1. 海洋生物 (halobios)	1. 海洋生物(halobios)	1. 海洋生物 (halobios)	1. 海洋生物 (halobios)
2. 海洋环境(marine environment)	2. 海洋环境(marine environment)	2. 海事(maritime)	2. 海洋环境(marine environment)
3. 海底(submarine)	3. 海洋学 (oceanography)	3. 海洋环境(marine environment)	3. 海底(submarine)
4. 极地 (polar region)	4. 海水(seawater)	4. 海水(seawater)	4. 大海(sea)
5. 海洋学 (oceanography)	5. 海洋资源 (ocean resources)	5. 深海(deep ocean)	5. 海水(seawater)
6. 生态(ecology)	6. 洋流(ocean current)	6. 大海(sea)	6. 海洋工程(marine engineering)
7. 深海(deep ocean)	7. 海床(seabed)	7. 海底(submarine)	7. 海洋学 (oceanography)
8. 大气(atmosphere)	8. 海洋工程(marine engineering)	8. 海洋学 (oceanography)	8. 海(sea)
9. 太平洋 (the Pacific Ocean)	9. 极地(polar region)	9. 大气(atmosphere)	9. 海床(seabed)
10. 海水(seawater)	10. 大陆架 (continental shelf)	10. 远洋 (distant ocean)	10. 深海 (deep ocean)

6 CONCLUSION AND FUTURE WORK

In this paper, the tripartite weighted graph with a weight assignment approach is proposed to model the relationship among words, characters, and sub-characters, and manage their semantics. Besides, with the tripartite weighted graph as the input, the Chinese word embedding model *insideCC* is designed to discover the semantic relation and interaction among different language components, strengthen the semantics of words with relevant sub-characters, and learn the embeddings of Chinese words. Furthermore, experimental results of word similarity and word analogy reasoning tasks verify that *insideCC* outperforms the state-of-the-art counterparts by a significant margin. Additionally, our methods are language-specific and can be employed to other languages with analogous reading and writing regularities to Chinese, such as Japanese.

Several improvements may be possible in our future work. First, since Tongyici Cilin only contains synonyms, we can incorporate other auxiliary dictionaries like Hownet [8] to discover and learn more accurate semantic relevance. Second, we can employ other models like the transformer [26] to optimize the ultimate semantic relevance. Finally, we can explore emerging methods to integrate the embeddings of multiple-granularity sub-characters.

7 ACKNOWLEDGEMENT

The research is supported by the National Natural Science Foundation of China under Grant Nos. 61932004 and 62072205.

REFERENCES

[1] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Mikolov Tomas. 2017. Enriching Word Vectors with Subword Information. In *Transactions of the Association for Computational Linguistics*, Vol. 5. 135–146.

[2] Shaosheng Cao, Wei Lu, Jun Zhou, and Xiaolong Li. 2019. cw2vec: Learning Chinese Word Embeddings with Stroke n-gram Information. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*. 5053–5061.

[3] Xinxiong Chen, Lei Xu, Zhiyuan Liu, Maosong Sun, and Huanbo Luan. 2015. Joint Learning of Character and Word Embeddings. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI 2015)*. 1236–1242.

[4] Zheng Chen and Keqi Hu. 2018. Radical Enhanced Chinese Word Embedding. In *Proceedings of the 17th China National Conference on Computational Linguistics*. 3–11.

[5] Lizhi Cheng, Weijia Jia, and Wenmian Yang. 2021. An Effective Non-Autoregressive Model for Spoken Language Understanding. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*. 241–250.

[6] Jacob Devlin, Mingwei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 4171–4186.

[7] Chinese Academy of Social Sciences Dictionary Department, Institute of Linguistics. 2012. *Modern Chinese Dictionary*. The Commercial Press.

[8] Zhendong Dong and Qiang Dong. 2003. Hownet-a hybrid language and knowledge resource. In *Proceedings of the IEEE International Conference on Natural Language Processing and Knowledge Engineering*. 820–824.

[9] Felix Hill, Roi Reichart, and Anna Korhonen. 2015. SimLex-999: Evaluating Semantic Models with (Genuine) Similarity Estimation. In *Computational Linguistics*, Vol. 41. 665–695.

[10] Zijing Ji, Xin Wang, Yuxin Shen, and Guozheng Rao. 2021. CANCN-BERT: A Joint Pre-Trained Language Model for Classical and Modern Chinese. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*. 3112–3116.

[11] Shen Li, Zhe Zhao, Renfen Hu, Wensi Li, Tao Liu, and Xiaoyong Du. 2018. Analogical Reasoning on Chinese Morphological and Semantic Relations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 138–143.

[12] Bin Liang, Rongdi Yin, Lin Gui, Jiachen Du, Yulan He, and Ruifeng Xu. 2021. Aspect-Invariant Sentiment Features Learning: Adversarial Multi-Task Learning for Aspect-Based Sentiment Analysis. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*. 825—-834.

[13] Yiyuan Liang, Wei Zhang, and Kehua Yang. 2018. Attention-Based Chinese Word Embedding. In *Proceedings of the 2018 International Conference on Cloud Computing and Security*, Vol. 11066. 277–287.

[14] Wentao Ma, Yiming Cui, Chenglei Si, Ting Liu, Shijin Wang, and Guoping Hu. 2020. CharBERT: Character-aware Pre-trained Language Model. In *Proceedings of the 28th International Conference on Computational Linguistics*. 39–50.

[15] Jiaju Mei, Yiming Zheng, Yunqi Gao, and Hungxiang Yin. 1984. *TongYiCiCiLin*. Shanghai: the Commercial Press.

[16] Yuxian Meng, Wei Wu, Fei Wang, Xiaoya Li, Ping Nie, Fan Yin, Muyu Li, Qinghong Han, Xiaofei Sun, and Jiwei Li. 2019. Glyce: Glyph-vectors for Chinese Character Representations. In *Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*.

[17] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. (2013). arXiv:1301.3781 [cs.CL]

[18] George A Miller. 1995. WordNet: a lexical database for English. In *Communications of the ACM*, Vol. 38. 39–41.

[19] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*. 1532–1543.

[20] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. 2227–2237.

[21] Yunqi Qiu, Kun Zhang, Yuanzhuo Wang, Xiaolong Jin, Long Bai, Saiping Guan, and Xueqi Cheng. 2020. Hierarchical Query Graph Generation for Complex Question Answering over Knowledge Graph. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*. 1285–1294.

[22] Yan Song, Shuming Shi, and Jing Li. 2018. Joint Learning Embeddings for Chinese Words and their Components via Ladder Structured Networks. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI-18)*. 4375–4381.

[23] Tzu-Ray Su and Hung-Yi Lee. 2017. Learning Chinese Word Representations From Glyphs Of Characters. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 264–273.

[24] Zijun Sun, Xiaoya Li, Xiaofei Sun, Yuxian Meng, and Jiwei Li. 2021. ChineseBERT: Chinese Pretraining Enhanced by Glyph and Pinyin Information. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 2065–2075.

[25] Ke Tu, Peng Cui, Daixin Wang, Zhiqiang Zhang, Jun Zhou, Yuan Qi, and Wenwu Zhu. 2021. Conditional Graph Attention Networks for Distilling and Refining Knowledge Graphs in Recommendation. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*. 1834–1843.

[26] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*. 5998–6008.

[27] Jian Xu, Jiawei Liu, Liangang Zhang, Zhengyu Li, and Huanhuan Chen. 2016. Improve Chinese Word Embeddings by Exploiting Internal Structure. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 1041–1050.

[28] Rongchao Yin, Quan Wang, Peng Li, Rui Li, and Bin Wang. 2016. Multi-Granularity Chinese Word Embedding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. 981–985.

[29] Jinxing Yu, Xun Jian, Hao Xin, and Yangqiu Song. 2017. Joint Embeddings of Chinese Words, Characters, and Fine-grained Subcharacter Components. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 286–291.

[30] Yun Zhang, Yongguo Liu, Jiajing Zhu, Ziqiang Zheng, and Shuangqing Zhai. 2019. Learning Chinese Word Embeddings from Stroke, Structure and Pinyin of Characters. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 1011–1020.

[31] Xinhua Zhu, Runcong Ma, Liu Sun, and Hongzhao Chen. 2016. Word Semantic Similarity Computation Based on Hownet and Cilin. In *Journal of Chinese Information Processing*, Vol. 30. 29–36.